NATIONAL SECURITY AGENCY (NSA) SYSTEMS AND NETWORK ATTACK CENTER (SNAC) SECURITY GUIDES VERSUS KNOWN WORMS

THESIS

Matthew W. Sullivan, 2d Lt, USAF
AFIT/GIA/ENG/05-07

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/GIA/ENG/05-07

NATIONAL SECURITY AGENCY (NSA) SYSTEMS AND NETWORK ATTACK CENTER (SNAC) SECURITY GUIDES VERSUS KNOWN WORMS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Information Assurance

Matthew W. Sullivan, BS

2d Lt, USAF

March 2005

AFIT/GIA/ENG/05-07

NATIONAL SECURITY AGENCY (NSA) SYSTEMS AND NETWORK ATTACK CENTER (SNAC) SECURITY GUIDES VERSUS KNOWN WORMS

Matthew W. Sullivan, BS

2d Lt, USAF

Approved:

/signed/

Rusty O. Baldwin, Ph.D. (Chairman)	Date

/signed/

Richard A. Raines, Ph.D. (Member)	Date

/signed/

Robert F. Mills, Ph.D. (Member)	Date

Acknowledgments

I want to thank the many people that helped me make it through the thesis process. First, I would like to thank Dr. Baldwin for the many hours of editing and guidance that he gave me. I would also like to thank Mr. Lacey for quickly providing me all the equipment and software that I needed as well as all of his technical support. Captain Chaboya also truly helped me out with his expertise in debugging and provided some great insight into the world of hackers. I also would like to thank my sponsors from the NSA for providing me with the CERT database of exploits. Last but not least, I would like to thank my wife, for putting up with the late hours and weekends that I put into this thesis.

Matthew W. Sullivan

Table of Contents

Page

List of Figures

Page

List of Tables

Page

AFIT/GIA/ENG/05-07

Abstract

Internet worms impact Internet security around the world even though there are many defenses to prevent the damage they inflict. The National Security Agency (NSA) Systems and Network Attack Center (SNAC) publishes in-depth configuration guides to protect networks from intrusion; however, the effectiveness of these guides in preventing the spread of worms hasn't been studied.

This thesis establishes how well the NSA SNAC guides protect against various worms and exploits compared to Microsoft patches alone. It also identifies the aspects of the configuration guidance that is most effective in the absence of patches and updates, against network worm and e-mail virus attacks. The results from this thesis show that the Microsoft patches and the NSA SNAC guides protect against all worms and exploits tested. The main difference is NSA SNAC guides protected as soon as they were applied where as the Microsoft patches needed to be written, distributed and applied in order to work. The NSA SNAC guides also provided protection by changing default permissions and passwords some worms and exploits use to exploit the computer as well as removed extraneous packages that could have undiscovered exploits.

NSA SNAC SECURITY GUIDES VS KNOWN WORMS

1 Introduction and Importance of Research Topic

Worms are similar to computer viruses in that they can destroy data on computers and networks, but they have the additional ability to spread and disrupt the network without human interaction. Worms have been spreading faster as Internet connectivity has increased, some worldwide in as little as 15 minutes. This gives little warning or time for defensive measures to be put in place. The research objective of this effort is to determine if the National Security Agency (NSA) Systems and Network Attack Center (SNAC) security guides, alone, are effective protection against worms and viruses.

Since the United States has become increasingly dependent on computer networks for both defense and commerce, small disruptions in these networks can cause both great distress and damage. Computer worms cost both money and man hours to correct wasting resources. Knowing how well or what parts of the NSA SNAC guides are effective can help to minimize the damage from worms and may protect systems in future attacks.

1.1 Outline of Research Goals

The goal of this thesis is to determine whether the National Security Agency (NSA) Systems and Network Attack Center (SNAC) security guides are effective protection against the infection and spread of worms. In addition, aspects of the configuration guidance that are most effective in the absence of patches and updates are identified.

Since Microsoft products are found on over 90% of desktop systems, 55% of servers [Thu03] and 53% of Fortune 1000 Internet web servers [Huc03], this research uses Microsoft based operating systems and worms that attack those systems.

Two LANs are used as a test bed, one with a default installation of the Windows Operating System and the other with varying levels of protection to determine how well the NSA SNAC guides protect the respective computers. The levels of protection are: initial setup, initial setup with current Microsoft patches installed, initial setup with only NSA SNAC guides applied, and initial setup and both current Microsoft patches installed and NSA SNAC guides applied. Worms are run against each of the levels of protection to determine which level of protection works best. These worms are selected based on whether they attack the operating system or applications.

1.2 Overview of Research Document

Chapter 2 is an introduction on the history of worms as well as an overview of how they work and some common attributes. It covers other ways to prevent worms from spreading, both host based and network based. An analysis on four worms, Code Red version I and II, Nimda, and SQL Slammer is also covered. The exploits tested in this thesis are also discussed. The chapter discusses current research on defeating worms.

Chapter 3 contains the methodology used to conduct the research. The goals are discussed as well as the approach to solve the problem. System boundaries, services, parameters and factors are presented as well. The experimental design and the evaluation technique are also covered.

Chapter 4 presents the results of the experiments. The chapter also examines reasons for the exploits failure or success with respect to the NSA SNAC guides. Several additional ways to secure the computers other than what the NSA SNAC guides suggest are examined.

Chapter 5 discusses of what type of configuration protects the best. What exploits the NSA SNAC guides protect against is specified. The significance of these findings to the security community is also given. Recommendations are made on how to better protect computer systems against worms.

2 Literature Review

In this chapter, Internet worms and the financial costs they have incurred are discussed. The operation of a worm is explained by describing common traits they all have. Detailed descriptions of four current worms are presented: Code Red & Code Red II, Nimda, and Sapphire. The exploits used in this thesis are also explained. Methods to prevent worms from attacking and destroying networks are also discussed. Since proper configuration of a computer is an effective way to stop worms, the National Security Agency's (NSA) System and Network Attack Center (SNAC) "how to" guides are described.

2.1 Worms

The United States has become increasingly dependent on computer networks for both defense and commerce and even small disruptions in this network can cause great distress among its users. Computer worms have the ability to disrupt the network without the human interaction that viruses require. Worms are stand alone programs that seek out vulnerable computers on the network wasting both computing time and bandwidth.

This chapter concentrates on worms and exploits written for Microsoft operating systems and products rather than on their UNIX counterparts for a number of reasons. The first reason is Microsoft products are found on over 90% of desktop systems, 55% of servers [Thu03] and 53% of Fortune 1000 Internet web servers [Huc03]. The fact that the Microsoft OS runs on a common x86 architecture while a Unix OS runs on numerous platforms allows worms to exploit more systems with a minimum of coding on part of the hacker. Furthermore, since most users of Microsoft products do so with no formal security training, they form the most vulnerable group to be threatened by worms. They also form a large group that, if combined, could form a large distributed denial of service attack. Furthermore, the NSA SNAC guides that are the subject of this research are targeted to the Windows-based platform.

2.2 Worms History and Cost

Worms predate the Internet; they are named after a 1975, John Brunner story, The Shockwave Rider [Arc99]. The major defining characteristic of a worm is they are self contained and require no interaction with a user to execute. This independent execution ability gives worms the ability to use a significant amount of network bandwidth. In early 1980 Xerox created user independent processes that were used as helpful services, but some were poorly written and demonstrated the future danger of worms when they continuously rebooted infected computers [Arc99]. The first self-replicating, self-propagating worm was created by Robert Morris Jr. as part of his doctoral work in 1988. The Morris worm shutdown the largest percentage of the Internet to date, nearly ten percent, and cost an estimated $10-100 million to cleanup [Sul98]. This damage was completely unintentional. Errors in the code caused computers to be infected multiple times, spawning new processes that eventually brought infected computers to a halt. The CERT Coordination Center, a federally funded center of Internet security expertise was formed as a direct result of the Morris worm’s ability to do so much harm in such a short period.

When the Morris worm was released, the Internet was largely homogeneous. This allowed the same worm to propagate throughout each server without alteration of the code. Until Windows became the dominate OS, the Internet had a variety of operating systems (OS) and platform architectures. Now, with the Microsoft OS controlling approximately 90% of desktop and 45% of servers, the Internet has returned to a relative state of homogeneity [Naz04].

Current worms usually take advantage of bugs and security holes to infiltrate networks. The SoBig and Blaster worms of 2003 resulted in the biggest cleanup and longest down time thus far. The SoBig.F worm alone cost over $30 billion for cleanup and according to experts; the Blaster worm may have contributed to the failure of the eastern US power grid on August 14th 2003 [AdG03].

About 70% of South Korean users access the Internet using broadband and in 2003 the SQL Slammer worm infected their top three Internet providers which virtually brought the Korean Internet to a halt [AdG03]. With the increase of broadband Internet connections to home users, worm damage and propagation is expected to increase substantially. Given their ability to cause damage, it is clear emerging worms need to be stopped before they spread. The first step is to find out how worms actually work.

2.3 How Worms Work

Since they do not rely on user interaction, worms are programmed with all the information they need to spread from the beginning. To speed up the process of creating worms, most hackers exploit published security flaws with readily available patches to gain access to computers. Some even use the published flaw’s code in their worm. Another type of worm, the so-called “zero-day” worms, are harder to prevent because they use vulnerabilities that haven’t been identified by the security community and don’t have patches. This makes it much more difficult to stop them. Even so, all worms share some basic characteristics: autonomy, replication, reconnaissance, attack, defense, command interface, and polymorphism [Tod03].

The first four characteristics, autonomy, replication, reconnaissance, and attack, are all present in modern worms. Autonomy is a fundamental ability in a worm since once unleashed, a worm should spread without intervention. Replication is also a key trait for a worm since it needs this to spread. Worms use reconnaissance to find other computers that have a vulnerability that can be exploited. When the worm attacks, it is usually done in a two stage process. First, the worm exploits the vulnerability and loads itself onto the computer, and then it executes code to start the process of replicating from that computer [Tod03].

Modern worms use the last three characteristics, defense, command interface, and polymorphism, to increase their destructive ability.   Modern worms use multiple attacks so they can exploit multiple vulnerabilities or different operating systems. In addition, they can take advantage of multiple vulnerabilities in multiple operating systems and report compromises to a central database. Once the worm loads itself on a vulnerable computer it must avoid detection using its defensive capabilities. It can change its process name to something obscure, like a critical system process. The worm could also disable detection systems or send decoy packets to make it hard to locate other infected computers. A worm may send an identical replica of its code or use polymorphism to send out a modified version. Worms use polymorphism so there is no single code signature to discover and block. This can be potentially devastating since many worm filters use signatures and are rendered ineffective if each instance is different.

There are four main reasons worms continue to be generated even though they produce a great deal of “noise” during intrusion; ease, penetration, coverage and persistence [Naz01]. Worms are easy to generate because automation makes tasks easier. Writing a worm is not necessarily fast. It can, in fact, take a long time. Worms can penetrate systems not only through effective code, but also through good fortune on the part of the attacker. Worms spread quickly due to their very nature; this coverage helps them persist over long periods of time since some users don’t patch their systems quickly or at all [Naz01].

Some of the exceptionally virulent worms like Code Red and Nimda have persisted on the network for months after patches have been applied since worm writers have targeted broadband users of late. Each of these reasons make worms a threat for the foreseeable future. The relative ease of writing a worm also ensures that they will be around for a while [Naz01].

2.4 Worms, Friend or Foe?

While most worms are used to cause damage, some worms have been used for productive tasks, albeit with mixed results. The Xerox worm of the early 1980’s operated at night to balance daytime processing load for large tasks or to update system files. Unfortunately, these worms had some unintended consequences and caused computers to continually restart showing that worms could be used for malicious actions [Arc99]. The recent Nachi / Welchia worm loaded itself onto vulnerable Windows machines and attempted to patch the computer so that neither it nor the Blaster worm could affect it anymore. While both of these worms tried to help automate tasks, there were serious problems with bandwidth utilization and unintentional system misconfigurations that could have serious consequences.

While all worms use network bandwidth and CPU time, some carry a payload to perform malicious actions on compromised computer such as installing a Trojan horse, keystroke logger, or other types of spy ware. Worms can also destroy files or other information unintentionally. Since even the best intentioned worms have been shown to cause problems therefore, it is best not to allow any worm to be transmitted across a network.

2.5 Future of Worms

Code Red and Nimda, which spread around the world in days, seem tame compared to the predictions concerning future worms. After the appearance of the Code Red worm, it was postulated that in the future worms could take over the Internet within 15 minutes [Naz04]. This type of worm was dubbed the Warhol worm. Today’s worms spread faster with less code, giving rise to the thought of the Warhol or flash worm. A flash worm could be achieved by scanning in advance for vulnerable machines and splitting up the worm distribution so the servers and network bandwidth is not overwhelmed [Naz04].

While today’s worms have been troublesome in both cost of cleanup and wasted bandwidth, the future looks even worse. To date, worms haven’t been overly malicious; they have mainly wasted bandwidth and caused temporary denial of service. Future worms could carry devastating packages that delete data causing widespread damage. This is especially true for broadband users, the new target among worms, since they seldom make backups of their data.

Worms frequently announce their terrorist or political agendas. The Code Red worm proclaimed, “Hacked by Chinese”, on the web pages it defaced. Future worms will likely try to spread messages of groups either by defacing web pages like the Code Red worm, causing some type of denial of service or worse [ArR01].

Future worms will also likely target new areas. Peer-to-peer networks, such as Kazaa and Bittorrent, encourage the swapping of files among users. These could be used spread worms through the exchange of tainted files. If embedded devices such as routers are attacked, entire networks could be taken off-line. Many other embedded devices, printers, home appliances, and broadband adapters, have become accessible to the web with little concern about their security vulnerabilities. Since these devices use firmware, upgrading them is difficult if not impossible making worms even more of a threat.

2.6 Case Studies

At this point, case studies of four particular worms are presented: Code Red & Code Red II, Nimda, and SQL Slammer.

2.6.1 Code Red & Code Red II (July 2001)

The Code Red worm uses a buffer overflow attack to gain access to Microsoft’s Internet Information Services (IIS) Indexing Service Dynamic Link Library (DLL) which had a known vulnerability at the time. A patch to fix the vulnerability had been released a month earlier. Code Red spawns 100 threads, one trying to alter the main web page and the other 99 trying to find new computers to attack [Naz04]. CERT describes the attack as a three step process. First, the worm tries to exploit a random computer with a buffer overflow on TCP port 80. Then, it changes the default web page on English language machines to read, “HELLO! Welcome to http://www.worm.com! Hacked By Chinese!” Finally, the worm performs one of three actions depending on the day of the month: propagate to other machines, flood a fixed IP address to create a denial of service attack, or sleep. Additionally, the IIS attack sometimes results in root level access to the compromised machine [CER02].

Code Red is one of the first worms to use the homogeneity of the Internet to spread with the same speed of the 1988 Morris worm. It also foreshadows information warfare with the politically motivated “Hacked by Chinese” slogan. Code Red was contained because of the flaws in its random number generator code and the ability to fool it into thinking a computer was already infected [Naz04]. Code Red 2 fixed the flaw in the random number generator resulting in a significant increase in the number of scans by the worm. The worm used TCP, so every instance of a Code Red worm had to wait for an explicit response from the computer it was attacking before it would continue which prevented it from spreading faster.

While Code Red II used the same buffer overflow of the original Code Red, it used a probabilistic island hopping approach instead of the less effective randomly generated IP address of its predecessor. This island hopping approach treats network blocks as islands and the worm focuses its attention on this local network before moving to another random destination network [Naz04]. It also creates an entry in the registry to flag the computer as compromised [CER01]. Finally, it generates backdoors on the compromised machines by loading the executables “cmd.exe” to executable script directories and a Trojan horse copy of “explore.exe” that maps the computer’s disk drives [Naz04].

2.6.2 Nimda (September 2001)

A little more than a month after Code Red II’s release Nimda was released. Nimda used the same probabilistic island hopping approach as Code Red II to infiltrate vulnerable servers. In contrast with other worms, Nimda uses multiple attack vectors to penetrate systems. In web server exploits, Nimda used backdoor shells from previously exploited Code Red II web servers and another exploit that allowed access of a computer’s true root directory and the execution of arbitrary programs [Naz04]. It also exploited a vulnerability in the Microsoft email client that automatically ran a MIME encoded readme.exe attachment [CER01a]. It spread using open network shares of MIME-encoded copies of itself that were automatically run if the preview option was enabled. Another web exploit uploads more exploits to an infected site. Since Nimda used many infection techniques, it has avoided complete removal and has remained largely active for many months after its first introduction to the network [Naz04].

2.6.3 SQL Slammer (January 2003)

SQL Slammer, also known as Sapphire and W32.Slammer, is the fastest spreading worm to date. Almost 90% of vulnerable computers were infected within 10 minutes on January 25, 2003, nearly an hour before anyone could even begin to protect against it [MPS03]. Five of the 13 root-name servers and huge sections of the Internet went off line in the first 15 minutes of a relentless packet storm. Sapphire used a buffer overflow attack on Microsoft SQL Server 2000 and Desktop Engine 2000 software. The vulnerability had been known for 6 months and a patch was available [CER03]. Due to improper software configurations, some victims didn't even realize SQL was running [Bou03]. The Sapphire worm infected nearly 75 thousand hosts and reached its maximum scanning rate in three minutes. At this point, network bandwidth limitations began to limit its spread. Sapphire also caused airline flight cancellations, interfered with elections, and ATM failures [MPS03]. This was the first worm to employ the concept of the Warhol worm. It was two orders of magnitude faster than Code Red. Luckily Sapphire didn't carry a malicious payload or the effects would have been much more severe [MPS03].

The Sapphire worm used a buffer overflow exploit that was contained in a single UDP packet, as opposed to the TCP scan of Code Red and Nimda. Since it used UDP, it didn't wait for a response and quickly consumed much of the available bandwidth.

> "Slammer's scanning technique is so aggressive that it quickly interferes with its own growth. Subsequent infections' contribution to its growth rate diminishes because those instances must compete with existing infections for scarce bandwidth. Thus, Slammer achieved its maximum Internet-wide scanning rate in minutes." [MPS03]

Fortunately, there were three problems with the Sapphire's random number generation code that helped limit the spread. Further, the Internet community was better trained to stop the spread of worms with the prior outbreaks of Code Red and Nimda and within an hour put in place UDP filters for 376-byte packets destined for port 1434

[CER03]. Additionally, port 1434 could easily be blocked. In contrast, blocking commonly used ports like 80 or 443 would effectively result in a denial of service that could have been catastrophic.

The disturbing aspect about this incident is the author of the Sapphire worm is described as only having decent programming skills. Much of the code taken was from the actual published exploit. This worm has now set the bar for future worms and is considered an alarming new standard. The fact that an average programmer can create the fastest spreading worm in history shows that automated defenses are a necessity since humans can't respond in nearly enough time to protect online resources [MPS03].

2.7 Types of Worm Preventions / Protection

While it may seem that Internet worms are invincible, there are many network and host-based techniques that are effective against them. The host-based approach has much finer control but the network approach is still needed to block the huge number of incoming packets that a worm can produce. While some of these methods require a great deal of preparation, they are well worth the effort when an especially rampant worm tries to invade a network. Active methods seek out and destroy worms.

2.7.1 Host-Based

There are many ways of preventing or slowing the spread of worms using a host-based approach including firewalls and anti-virus software. A host-based firewall is a great tool to prevent the spread of a worm that breached the larger network firewall. Firewalls however, cannot block worms through ports that must remain open. Anti-virus software can get rid of worms on a machine, but requires constant updates on worm signatures. Another problem with host-based firewalls and anti-virus software is the amount of time to setup [Naz04]. There are also potential problems with polymorphic worms that change their signatures or quickly propagating worms that could overwhelm these tools.

Other ways of preventing worms is to lower the privileges on software or to use sandboxing or cratering. If software is running at root level, any compromise could result in a worm gaining that level of privilege; therefore running a process at a lower level would require extra steps be taken for the worm to compromise a system. Sandboxing is another way of controlling worms. Sandboxing runs processes in a restricted region. While in this region, the worm is unable to elevate privileges or alter files outside of the region. Experts agree that sandboxing is too resource intensive to be used effectively [Rob04]. Another novel way to stop worms is through cratering. Changing access control lists for certain files a worm requires to run would render it ineffective [Lie03]. This solution was used in the 1988 HI.com worm where experts recommended creating a file of the same name without read or write access [Naz04].

Misconfiguration of software seems to be one of the leading ways that worms exploit a system. Many software packages install unneeded routines by default. Systems can be made more secure by reducing the number of services offered. Most worms exploit vulnerabilities that have patches available. By installing current patches, worms would not be able to gain access to a system. Furthermore, most worms are released within 1 month of the patch’s release [Naz04]. Proactively scanning a network to determine what services are offered on ports and installing patches for those port services is a good practice. Installing the latest patches is, however, could cause downtime and the patch could be incompatible with already installed software.

Another prevention technique observes host behavior to determine if it has been compromised. There is a high learning curve with this method since it must be customized to a particular network, but it can limit an infected host from spreading the worm any further. The problem with this solution is it won’t stop passive worms, or worms that spread using the current usage patterns of the network [Naz04].

2.7.2 Network-Based Solutions

Network solutions should be used in conjunction with host-based solutions to form a better defense against worm based attacks. Network solutions depend on both perimeter and subnet firewalls and on intrusion detections systems. These can be used alone or can be integrated for better protection [Naz04].

Perimeter firewalls prevent a worm from penetrating the network outer layer thereby protecting the intranet resources. It can also prevent a worm leaving an affected network. Subnet firewalls add an additional layer of protection in case the worm passes through the perimeter firewall. While firewalls can’t ensure a network can be accessed by computers outside the firewall, they can protect the network behind the firewalls perimeter.

An Intrusion Detection System (IDS) can detect worms. IDS that create rules for network firewalls could prevent worms in their initial stages. However, firewalls could become overloaded with rules causing a denial of service [Naz04].

A hardware solution called the Field-programmable Port Extender (FPX) scans 2.4 billion bits per second and drops any data deemed malicious [LMK04]. While this throughput is a sizable increase from traditional software firewalls, worm signatures must be constantly updated for it to be effective.

TCP worms can be stopped by the LeBrea program. LeBrea looks for worms trying to connect to unused IP addresses on a network. The worm is “fooled” by completing the TCP three-way handshake only to put the worm computer “on hold” by keeping the connection open indefinitely. This virtually halts the worm by having its outgoing connections idle instead of looking for other hosts to infect [Lis03].

2.7.3 Other Protections

While host-based and network based protections counter worms passively, other methods seek out worms and their networks to destroy them. These methods are controversial and legally questionable because they search through intranets much like the worm they are trying to fight. Some can also cause a high load on a network that is already under strain from the spread of the worm.

Some active attacks against spreading worms send messages to the infected machine to shut down using the same attack as the worm. This slows the spread of a worm by shutting down machines that are replicating worms. When a worm initiates a check to see if it has already infected a machine, another active approach sends out a false message to the worm that the computer has already been infected. This approach is quite time consuming depending on the number of computers in the network [Naz04].

Some worms use a central location to update their code. To attack the worm's host network itself, an inoperable module could be installed at this central update node. This inoperable module would spread to newly infected nodes stopping the worm in its tracks. However, worm writers could easily defeat this by using public key encryption to update the module [Naz04].

Another way to stop worms is to send out a worm to patch computers. Worms like Bagle and Netsky each install themselves and uninstall the other [Rob04]. The Welchia worm downloads Microsoft updates and attempts to unload the Blaster worm [Sym04]. Many factors must be considered when writing this type of worm. If this "good" worm has errors, it may cause a bigger problem than the original worm. The bandwidth that the new worm uses compounds the problem with a potential denial of service. Finally, it is illegal to have a worm "fix" other computers just as it is for the hacker to release the first malicious one [Naz04].

2.8 NSA SNAC Guides

In 2001, a Congressional oversight committee learned that over 155 separate government computer systems had been hacked. This led to the enforcement of some established policies such as the Computer Security Act of 1987 which had a dual purpose. The first was to create a set of minimal security practices for Federal computer systems that contain sensitive information. The second was to assign responsibility for developing standards and guidelines to the National Institute of Standards and Technology (NIST) with guidance from the National Security Agency (NSA) [Cor01].

Many security vulnerabilities can be fixed by simply configuring the system properly. The NSA, working with Defense Information Systems Agency (DISA), NIST, FBI, SANS Institute, Center for Internet Security and other vendors, have developed a set of benchmark security configuration guides to provide a “pre-flight checklist” of security settings [Wol03].

The NSA has recently de-classified a group of documents it created to secure the Microsoft and some UNIX operating systems and applications [NSA04]. These NSA Systems and Network Attack Center (SNAC) guides use a top-down approach to securing a computer and are broken into six broad categories: Application Guides, Database Server Guides, Operating System guides, Router Guides, Supporting Document Guides, and Web Servers Guides. The NSA SNAC guides have in-depth explanations of how to secure their respective category as well as detailed instructions on how to perform those actions. The checklists at the end of the chapters are to the point and allow system administrators with an in-depth knowledge of their systems to setup the computers quickly. An example of a checklist entry from the *Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0* [Wal02] is:

- Remove all NTFS permissions from the Inetpub directory, and assign only required access groups and accounts (i.e., remove everyone, add WebUsers, WebAdmins, etc.)
- Establish logical directory structure (i.e., separate static content, html, asp, scripts, executables into different labeled directories)
- Set NTFS permission on directory structures as required
- Delete/move all sample directories and scripts that execute the samples

The NSA’s 60 Minute Network Security Guide [NSA02], part of the Supporting Document guides section, provides an overview of security in both the Windows and

UNIX environments. This guide defines the properties that make a good security policy. The most important aspect of a good policy is to have buy-in from all involved which ensures both the writers of the policy and those who implement the policy agree. The policy must have guidelines for implementation and be enforced with appropriate security tools [NSA02].

2.9 Exploits

The exploits used in this thesis are now discussed. Each exploit was selected to test the ability to compromise the OS and selected services used.

2.9.1 OS Exploits

Worms like Blaster (August, 2003) and Sasser (April, 2004) send out random IP addresses which make it difficult to use them to attack other computers without extensive modification to the worm. Instead of using these worms, the actual exploit that the respective worms employed is used.

2.9.1.1 DCOM RPC Exploit

The Windows Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) buffer overflow exploit is used by worms like MSBlaster. This exploit is described in the Microsoft Security Bulletin MS03-026 originally posted on 16 July, 2003 [Mic03]. An attacker can send a buffer overflow to ports 135, 139, 445 or other RPC configured ports and gain system privileges for remote code execution. These ports are not intended to be used in a hostile environment are normally blocked with either a hardware firewall or a software firewall such as Windows Internet Connection Firewall (ICF) that is built into Windows XP Professional.

This experiment used the DCOM exploit written by Moore and analyzed by Wayne J Freeman [Fre03] which sends the buffer overflow to port 135 where the RPC improperly checks it. It then allows this malformed message to overflow the DCOM process and open a command shell on port 4444 with system level privileges.

2.9.1.2 LSASS Exploit

The Local Security Authority Service (LSASS) Buffer Overflow buffer overflow exploit is used by worms like Sasser, Korgo, Phatbot, Donk and Bobax. This vulnerability is described in Microsoft Security Bulletin MS04-11 [Mic04]. This exploit attacks certain Active Directory service functions in LSASRV.DLL with a buffer overflow that causes the DsRolerUpgradeDownlevelServer function to write entries to the dcpromo.log file. It also lets the attacker remotely execute code of their choosing. This exploit was discovered by eEye Digital Security and uses code written by Houseofdabus and analyzed by Travis Abrams [Abr04]. It tries to connect to port 445 remotely and opens a port of your choosing on the vulnerable computer.

2.9.2 Microsoft IIS Extended Unicode Directory Traversal Vulnerability

The Unicode directory traversal exploit, as discussed in Microsoft Security Bulletin MS00-078, is used by worms like Nimda. This exploit allows attackers to move out of the web root directory and access any file with the basic Internet user permissions by replacing the forward or backward slash with its respective UNICODE character.

2.9.3 Outlook Exploit

To test Internet Explorer 6.0 on the Microsoft XP Professional computers, Georgi Guninski’s security advisory #49 [Gun01] is used. This exploit uses Active X to control

"Microsoft Outlook View Control" which permits access and manipulation of the user's mail messages through Internet Explorer. It also allows the execution of arbitrary programs through Outlook's Application object.

2.9.4 Multipurpose Internet Mail Extensions (MIME) Header Exploit

The MIME type exploit as described in the Microsoft Security Bulletin MS01-020 [Mic01] is used by worms such as Klez, Bugbear, Mydoom, and Sobig. The original code that proved that this concept would work was written by Juan Carlos Garcia Cuartango. Microsoft Internet Explorer uses MIME to extend the functionality of Internet mail to allow formats other than just ASCII text to be used. MIME headers are used only to evaluate if the embedded file is potentially dangerous and not when the file is actually processed on the computer. When the embedded file is misrepresented it could allow potentially dangerous code to be processed on the vulnerable computer with the permissions of the current user [Mic01].

2.10 Summary

This chapter covered the background on Internet worms as well as the financial costs that have resulted. Some common traits of worms are described as well as what the future holds for worms. Four worms were covered in detail: Code Red & Code Red II, Nimda, and Sapphire. Each exploit used in the thesis is also discussed. Ways to stop worms from disrupting the network as well as the NSA SNAC guides were described.

3 Methodology

This chapter covers the goals and hypothesis of this thesis. It also covers the approach taken as well as the system boundaries.

3.1 Goals and Hypothesis

The intent of this thesis is to determine how well the NSA SNAC security guides protect the Windows 2000 Server and Windows XP Professional Workstation operating systems. It also looks at protection of the following applications from selected Windows-based worms and exploits: Internet Information Services (IIS) 5.0, SQL Server 2000, and Exchange 2000/ Outlook 2002/ Outlook Express. This experiment also determines how well the NSA SNAC guides protect against worms on a newly installed operating system (OS) and applications with and without recommended patches.

It is expected that Microsoft patches protect against most of the chosen worms and exploits since they are written specifically to stop them. It is unknown how well the SNAC guides protect an initial setup and no patches.

3.2 Approach

The effectiveness of the SNAC guides is evaluated using two LANs connected by a Cisco 2600 router as shown in Figure 1. In place of certain worms, the actual exploit is used because of the randomness of their connections to other nodes.

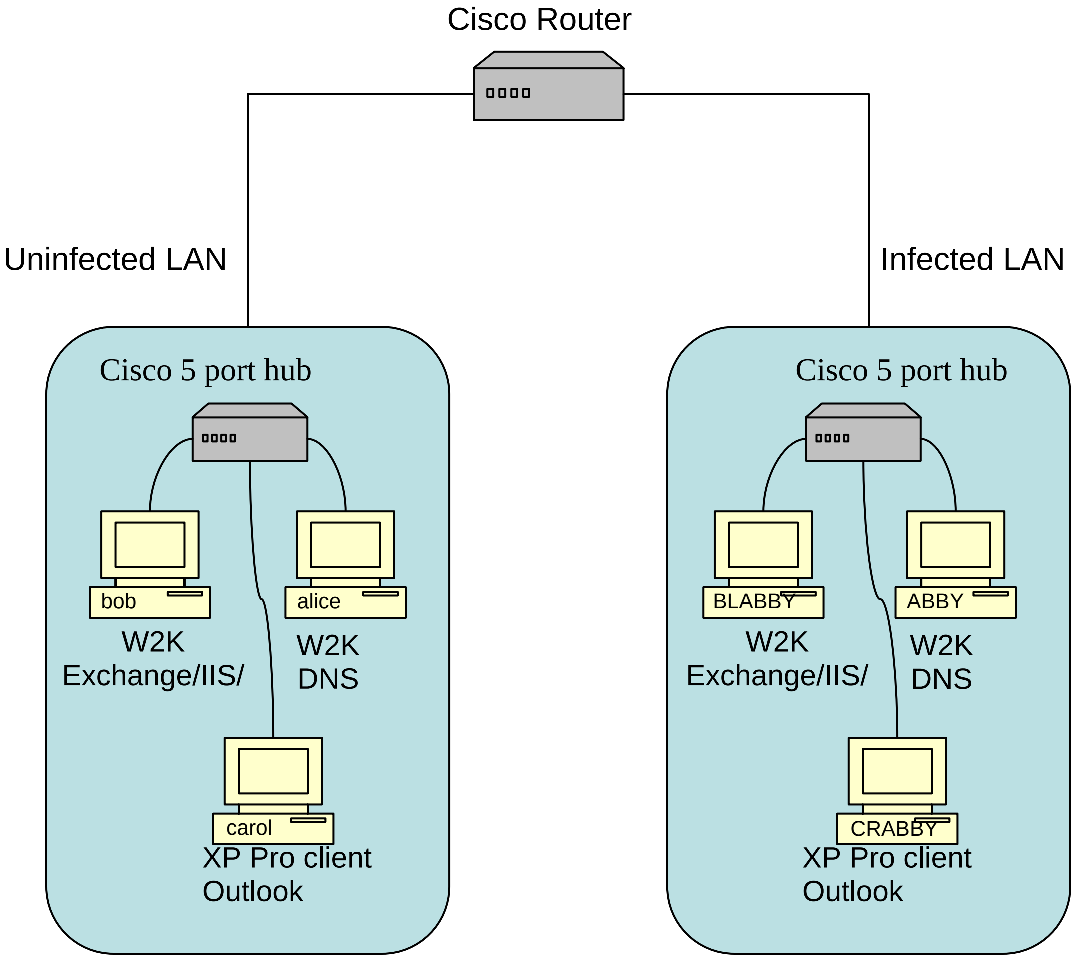

Figure 1: System Including IDS

One LAN is the Infected LAN and serves as a launching point for the worm or actual exploit. This LAN only used the initial setup of the Microsoft OS and applications in order to make sure that the worms can propagate without hindrance. The other (initially) Uninfected LAN is used to determine how well the NSA SNAC guides protect against worm infection using four configurations:

1) Initial install from Microsoft CDs

2) Initial install and all current patches from Microsoft Update website installed

3) Initial install and NSA SNAC guides incorporated, no patches are installed except Service Pack 1 which is required to install Exchange 2000

4) Initial install, all Microsoft patches and NSA SNAC guides incorporated

3.3 System Boundaries

The system under test (SUT) is called the Worm Protection System and includes computers with Windows 2000 Server operating system, and computers with Windows XP Professional (see Figure 2).

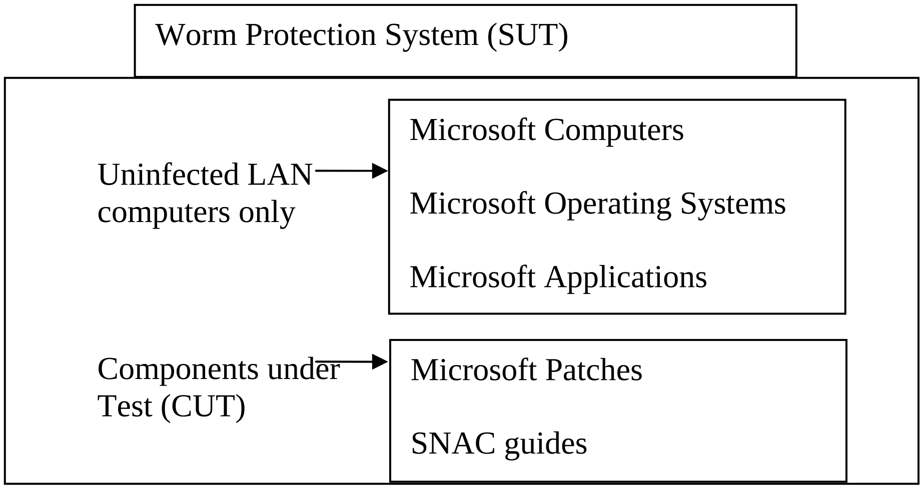
Worm Protection System (SUT)
Uninfected LAN computers only
Microsoft Computers
Microsoft Operating Systems
Microsoft Applications
Components under Test (CUT)
Microsoft Patches
SNAC guides

Figure 2: System under Test (SUT)

It also includes the following Microsoft applications; Internet Information Services (IIS) 5.0, SQL Server 2000, and Exchange 2000/Outlook 2002 / Outlook Express. The components under test are the NSA SNAC security guideline settings and all the current Microsoft patches for these applications as well as those for the OS. The scope of this experiment is limited to using the NSA SNAC guides and current patches only, no other means of preventing worms, such as firewalls or packet filtering, are used.

3.4 System Services

This system provides one service: protection against network propagated worms and exploits. There are two possible outcomes of this service; system is vulnerable or system is not vulnerable. A system is vulnerable when a worm or exploit has executed its particular attack vector and has compromised the intended service on the target computer. A system is not vulnerable when the worm is unable to compromise the intended service. This research does not examine denial of service attacks.

3.5 Workload

The workload in this research consists of selected Windows-based worms; namely, versions of the CodeRed worms and the Slammer worm from the CERT/CC Artifact Catalog [build 528]. The following exploits are used. Unicode Web Traversal [Sec00], the Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) exploit [Fre03], the Local Security Authority Subsystem Service (LSASS) exploit [Abr04], the MIME exploit described in Microsoft Security Bulletin (MSB) MS01-020 [Mic01], and the Outlook XP exploits described in Georgi Guninski's security advisory #49 [Gun01]. These worms and exploits are selected to test the ability to compromise the Windows OS and selected services, IIS 5.0, SQL Server, and Exchange 2000/Outlook 2002, while the NSA SNAC guides are designed to protect.

3.6 Performance Metrics

The performance metrics are based on whether or not the computer is vulnerable to the exploit / worm. The outcome is either system is vulnerable or system is not vulnerable. Since this experiment only tests whether a particular system is vulnerable from exploits against the specific vector of attack, there is no collection of other data such as how rapidly the worm spreads or how much bandwidth it used.

3.7 Parameters

The system parameters for this experiment are listed below:

- Computer Setup: Each is loaded with an OS, Windows 2000 version 5.0.2195 or Windows XP Pro version 5.1.2600, and the appropriate applications; Active Directory/DNS, IIS 5.0, SQL Server 2000, and Exchange 2000/Outlook 2002
- Number of Computers: There are three computers on both the Infected and the Uninfected LAN. These are used to simulate an actual working environment with a Windows 2000 DNS server, a Windows 2000 Exchange / IIS / SQL server and a Windows XP Professional client computer.
- Security Setup: The Infected LAN has an initial setup while the Uninfected LAN has four security configurations.
- Worm / exploit entry points: worms and exploits are released from the Infected LAN using the standard method of deployment explained in Chapter 4.

The worm / exploit workload parameters for this experiment are:

- Worm / exploit target of attack: OS and/or applications

3.8 Factors

The system factors and corresponding values for this experiment are:

- Security Configuration Setup:

    1) Initial install from Microsoft CDs

    2) Initial install and all current patches from Microsoft Update website

    3) Initial install, all Microsoft patches and NSA SNAC guides

    4) Initial install with NSA SNAC guides incorporated (no patches are installed except Service Pack 1 which is required to install Exchange 2000). The following NSA SNAC guides are used in the configuration of the computers:

Guide to Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services [Chr01]
Guide to Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services (Checklist Format) [Chr01a]
Guide to Securing Microsoft Windows 2000 Group Policy [Han01]
Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set [Han01a]
Guide to Securing Microsoft Windows 2000 Active Directory [SaR00]
Guide to Securing Microsoft Windows 2000 DNS [Ste01]
Guide to Securing Microsoft Windows 2000 File and Disk Resources [HaM02]
Guide to the Secure Configuration and Administration of Microsoft Exchange 2000 Version 1.2 [Pit03]
Guide to Secure Configuration and Administration of Microsoft Internet Information Services 5.0 [Wal02]
Guide to Secure Configuration and Administration of Microsoft SQL Server 2000 [ChH00]
Guide to Securing Microsoft Windows XP [BCH03]
Guide to Securing Microsoft Internet Explorer 5.5 Using Group Policy [Doe02]

The one divergence with the security setup is that the NSA SNAC guides call for all recent Microsoft patches to be installed. Only Service Pack 1 is applied to the initial setup with NSA SNAC guides to test how well the NSA SNAC guides work alone without Microsoft patches.

The workload factors are:

- Worm attack vectors, worms are selected to attack the following categories and tested against each level of security setup:
    - Operating Systems, Windows 2000 and Windows XP Pro:
        - DCOM RPC exploit [Fre03], LSASS exploit [Abr04]
    - IIS 5.0:
        - Multiple Code Red worms from CERT/CC Artifact Catalog, Unicode Web Traversal [Sec00]
    - Exchange / Outlook XP
        - the MIME exploit described in MSB MS01-020 [Mic01], the Outlook XP exploits [Gun01]
    - SQL Server
        - Slammer worm from CERT/CC Artifact Catalog

3.9 Evaluation Technique

The hypothesis is tested by direct measurement of a real network. Currently there are no simulations that can directly model the vulnerabilities and their subsequent fixes with patches. Validation of the results is done by examining the computer for evidence of infection based on known results of an attack.

Validation is performed on the worm or exploit on each node of the network. Every worm /exploit is run on an initial setup to make sure that it functions as expected.

Each computer node is checked to make sure it is setup correctly in each configuration. The network is checked to make sure that it is sending and receiving packets correctly. Ethereal is used on each machine to verify that a specific worm is working correctly and that it traveled across the network.

3.10 Experimental Design

The experimental design for this research is a full factorial design with replications. This allows for the examination of every possible combination of workloads and configurations. The number of factors, levels, and replications:

- Number of computer configuration setups = 4

    1) Initial install from Microsoft CDs
    2) Initial install and all current patches from Microsoft Update website
    3) Initial install, all Microsoft patches and NSA SNAC guides
    4) Initial install with NSA SNAC guides incorporated (no patches are installed except Service Pack 1 which is required to install Exchange 2000). The following NSA SNAC guides are used in the configuration of the computers:

- Number of replications = 2

    The second replication is done to verify that the results are the same.

- Number of worm workloads (the number of computers on a LAN represent the number of computers that are susceptible to the particular exploit):

    - DCOM RPC exploit = 3 computers on LAN * 4 setups = 12
    - LSASS exploit = 3 computers on LAN * 4 setups = 12
    - Unicode Web Traversal = 1 computer on LAN * 4 setups = 4
    - MS01-020 exploit = 1 computer on LAN * 4 setups = 4
    - Georgi Guninski’s exploit= 1 computer on LAN * 4 setups = 4
    - Slammer worm = 1 computer on LAN * 4 setups = 4
    - Code Red versions = 1 computer on LAN * 4 setups = 4

Total number of experiments (4) * (2) * (44) = 352

3.11 Summary

The experiments outlined in this chapter determine how well NSA SNAC guides protect against specific worms and exploits compared to an initial setup or patched systems. The system boundaries are outlined as the computers involved including their OS and applications, the NSA SNAC guides and current patches.

It is expected that current Microsoft patches block worms and exploits better than the NSA SNAC guides since they are written specifically for them. The experiments performed and the data received from these experiments is discussed in the next chapter.

4 Results

This chapter introduces each type of exploit and the results obtained during the exploit. Each exploit is presented in Chapter 2 and the results are described below for each configuration. Some alternative ways to protect against the exploit are also discussed here. A short conclusion of each exploit is also provided in each section.

4.1 Operating System Exploit Results

4.1.1 DCOM RPC Exploit

Windows 2000 Server

The DCOM exploit is successful on the initial configuration of Windows 2000 Server opening a command prompt to "C:\WINNT\System32". The exploit failed on all other configurations with the exception of the initial install with NSA SNAC guides when Internet Protocol Security (IPSec) is turned off. When IPSec is used to block the ports that are vulnerable the exploit is unsuccessful.

Security Focus [Sec03] states that another way to protect from the exploit is by having the Distributed COM be turned off, but states that this can only be done on Windows 2000 with Service Pack 3 installed. The problem with this solution is that it could create problems with the communication between the Active Directory /DNS server and the Exchange Server, which are closely linked and need to communicate on these ports.

Windows XP Professional

The Windows XP Pro initial setup is also compromised by the exploit opening a command prompt to "C:\Windows\System32". The exploit failed on all other configurations including the initial configuration with NSA SNAC guides when the built-in ICF enabled. When the ICF is disabled, the DCOM RPC exploit is successful on this configuration.

4.1.2 LSASS Exploit

Windows 2000 Server

The Travis Abrams experiment used a Windows 2000 computer with Service Pack 3, whereas this experiment used Service Pack 1 on both the initial and initial with NSA SNAC guides and Service Pack 4 on the patched configurations. With only Service Pack 1 installed the LSASS exploit restarts the computer after connecting.

The LSASS exploit failed on the initial setup on the Active Directory Windows 2000 Server, but succeeded on the Exchange / IIS / SQL Server. The exploit failed on all other Windows 2000 configurations. The exploit didn’t work on the initial setup of Windows 2000 Server with only NSA SNAC guides applied because the Local Security Policy “Additional restrictions for anonymous connections” setting is set to “No access without explicit anonymous permissions.” This prevented the LSASS exploit from connecting to the NetBIOS null session.

Security Focus recommends creating a read-only ‘dcpromo.log’ to stop this vulnerability [Sec04] which is why the exploit failed on the initial setup on Windows 2000 Active Directory / DNS server. They also recommend TCP/IP filtering to block all un-initiated inbound TCP traffic to any port. TCP/IP filtering may cause problems with the interaction of the Active Directory /DNS server and the Exchange Server which need to communicate over this port. Another approach is to stop the server service; unfortunately this is needed for IIS and Exchange administration to function correctly.

Windows XP Professional

The LSASS exploit succeeded on the Windows XP Pro computer with just an initial setup. The exploit failed on all other configurations. When the ICF is disabled on the initial configuration with NSA SNAC guides, the exploit succeeded. The exploit succeeded even when the two Local Security Policies: “Network Access: Do not allow anonymous enumeration of SAM accounts” and “Network Access: Do not allow anonymous enumeration of SAM accounts and shares” are enabled. A reason for this may be that the “Restrict Anonymous = 2” is no longer a valid setting for Windows XP Professional which is present in Windows 2000 Server. This setting fully prevents enumeration of the users and shares [Cer02].

4.1.3 Operating System Exploit Summary

The configurations with patches protected the computers since these patches are written specifically for the exploit. Note that all these patches were written after the exploit was discovered. The patches made it possible to prevent the buffer overflows by altering the vulnerable code.

The NSA SNAC guides could not prevent inherent buffer overflow exploits to the Operating System, but with IPSec enabled it could prevent the packets from getting to the computer. IPSec could also prevent an insider threat from attacking these NetBIOS ports, which are usually open behind a firewall.

4.2 IIS Exploit Results

4.2.1 Microsoft IIS Extended Unicode Directory Traversal Vulnerability

The exploit from Security Focus [Sec00] is used on the IIS 5.0 server on all configurations. The initial configuration is vulnerable to this exploit, while all other configurations are found to be secure.

4.2.2 Code Red Worm

The actual Code Red worm binaries, ‘codered.D’, ‘red1.bin’, ‘red2a.bin’, ‘red2b.bin’, from the CERT/CC Artifact Catalog [build 528] database are used to test the IIS 5.0 server. The Code Red worm exploit sends a buffer overflow to the Indexing Service DLL. Code Red exploited the ‘Idq.dll’ file because the script mappings for the Internet Data Query (.idq) and Internet Data Administration (.ida) files are present.

In this exploit, the binaries are sent to the Uninfected IIS server with NetCat on port 80. The initial configuration is vulnerable to each binary when tested; all other configurations prevented the exploit from working.

4.2.3 Conclusions

The NSA SNAC guides changed the IIS home directory so that it is on a drive separate from the operating system preventing the UNICODE traversal. The guides also rename common directories and eliminating unnecessary ones in case any of these are vulnerable. The NTFS file permissions are also changed so that minimal permissions are granted and that “Guest” and “Everyone” are removed from the IIS directories. This prevents the “IUSR” account from having too much control over the IIS directories.

The NSA SNAC guides also remove any unneeded script mappings to prevent any potential vulnerability that these “.dll” files could have from affecting the security of the web server.

4.3 SQL Server Exploit Results

4.3.1 SQL Slammer Worm

To test the Microsoft SQL Server 2000, the SQL Slammer ‘worm.bin’ binary from CERT/CC Artifact Catalog [build 528] database is used. The SQL Slammer worm uses a buffer overflow against SQL Server 2000 as described in Chapter 2. The SQL binary is sent to the SQL Servers in each configuration using Netcat.

This exploit is successful on the initial setup, but is unsuccessful on all other configurations.

4.3.2 Conclusions

The NSA SNAC guides recommend the use of Windows Authentication Mode. This prevented the worm from connecting to the server. Also, by changing the port like the NSA SNAC guides suggest, it would be more difficult and require more coding for the worm to find and try to exploit. In addition, the NSA SNAC guides recommend using IPSec to secure the server. This experiment didn’t use IPSec, but it would certainly add another substantial layer to the security of the SQL server as shown by the success of IPSec in the Operating System exploits.

4.4 Internet Explorer (IE) / Email Exploits

4.4.1 Microsoft IE MIME Header Exploit Results

The MIME type exploit as described in the Microsoft Security Bulletin MS01-020 [Mic01] is used by worms such as Klez, Bugbear, Mydoom, and Sobig. The original code that proved that this concept would work was written by Juan Carlos Garcia Cuartango. The demonstration from Inside Security is used to test the Windows 2000 servers with Internet Explorer 5.0 [Ins01]. Since Internet Explorer 6.0, installed by default on the Windows XP Professional is not affected, it was not tested. This exploit has an incorrectly configured MIME map on the server and allows "foo.vbs" to run on the client which writes a test.txt file to the C: drive.

The initial configuration is vulnerable to this exploit and had the "test.txt" file written to the C: drive. The patched Windows 2000 configuration had Internet Explorer 6.0 so it is not vulnerable to the exploit. The NSA SNAC guides configuration is vulnerable to this exploit when the security setting "file download" is enabled but when "file download" is disabled the exploit failed to run the script which prevented the creation of the "test.txt" file to the C: drive.

4.4.2 Microsoft IE / Outlook Exploit Results

The initial configuration of this exploit deleted email from the user's Outlook as well as opened a command prompt that is able to execute any command. The patched system opened the Outlook mail in Internet Explorer, but it didn't delete the mail or open up the command window. The NSA SNAC guides are not vulnerable with Active X disabled and didn't open Outlook emails or the command prompt.

4.4.3 Conclusions

The patched configuration does nothing to disable Active X or File Download which could lead to other exploits. They do however protect from both of these exploits although they still allow Internet Explorer to access Outlook's Application object. The NSA SNAC guides let the system administrators choose to enable Active X and File Download based on usability in the Internet Zone. While this is done to ensure functionality for end users, these tests show it is a risk to keep them enabled.

4.5 Summary

This chapter covers all the results of the exploits on each experiment conducted. It also explained the exploits and how the NSA SNAC guides protected against them. Some alternative protection methods are also covered. Table 1 identifies the results of how the four different security setups performed against each exploit or worm. The Initial system configuration is vulnerable to all exploits. The NSA SNAC guides configuration as well as the patched system prevented the attacks.

Table 1: Result of Exploit on Different Configurations

	Type Configuration			
Exploit/worm	Initial	Initial + NSA SNAC guides	Initial + Patches	Initial + Patches + NSA SNAC guides
DCOM RPC	System Vulnerable	Exploit Failed w/ IPSEC or XP firewall System Vulnerable w/o IPSEC or XP firewall	Exploit failed	Exploit failed
LSASS	System Vulnerable	Exploit failed	Exploit failed	Exploit failed
Code Red	System Vulnerable	Exploit failed	Exploit failed	Exploit failed
Unicode Traversal	System Vulnerable	Exploit failed	Exploit failed	Exploit failed
SQL Slammer	System Vulnerable	Exploit failed	Exploit failed	Exploit failed
Georgi Guninski's security advisory	System Vulnerable	Exploit failed w/ Active X disabled	System Partially Vulnerable	Exploit failed w/ Active X disabled
MS01-020 (on IE 5.0)	System Vulnerable	Exploit failed when file download disabled	Exploit failed	Exploit failed

5 Conclusions

This chapter presents the conclusions of the research. It compares the results of all configurations as well as gives the reason for the results on the configuration using the NSA SNAC guides.

5.1 Conclusions of Research

While both the NSA SNAC guides and the Microsoft patches are comparable in their protection against the exploits, as shown in Table 1, there are many factors to look at when trying to determine what type of configuration is better. The most important factor to consider is from what point the exploit is discovered to the time when the system is protected. Another issue is what type of vulnerability the exploit is attacking.

While the patched configuration protects about as well as the NSA SNAC guides configuration, there is a big difference in the timeliness of the fix to the vulnerability. The NSA SNAC guides are applied to the initial configuration so the computers are protected as soon as they are put online. The patched systems, on the other hand, are vulnerable until the patch for the particular exploit is released and then installed on the computers.

Furthermore, patches on the computers do not secure passwords, change security settings, limit access or remove extraneous packages that could have undiscovered exploits. Some exploits rely on weak passwords which patches do not fix. The NSA SNAC guides make sure that the passwords meet complexity requirements as well as being 12 characters long.

The NSA SNAC guides limit which ports that can be accessed by using IPSec or XP Professional’s built-in firewall. This not only stops the known buffer overflow vulnerabilities, but could potentially stop any new exploits from attacking these ports. It also can prevent insider threats since many organizations’ NetBIOS ports are open behind their firewall.

The NSA SNAC guides also protect applications with ports that can’t be closed, like IIS, and SQL Server. The NSA SNAC guides recommended removing superfluous Internet Server Application Program Interface (ISAPI) filters as well as unused sample directories from IIS to prevent exploits. They also recommended the web root directory be on a separate drive from the OS to prevent UNICODE traversal exploits. The SQL Server should be moved to a non-standard port which would help against worms that scan for the standard port. Further, the NSA SNAC guides recommend Windows Authentication Mode for the SQL Server which uses the built in security authentication of the Windows OS. Another recommendation is to use IPSec on these SQL services so connections to your computer are limited thus reducing your exposure to possible exploits. The NSA SNAC guides also disable unneeded services to prevent exploitation. This further reduces what ports are listening and services that could be vulnerable.

With all these facts and the results of the experiments it is reasonably certain that the NSA SNAC guides provide better protection than Microsoft patches alone. Specific reasons that the NSA SNAC guides prevented the exploits from working are shown in Table 2.

Table 2: NSA SNAC Guides Configuration Results

Exploit/worm	Reason exploit failed on initial configuration with NSA SNAC guides
DCOM RPC	Windows 2000: IPSec blocked vulnerable ports XP Professional: ICF blocked vulnerable ports
LSASS	Windows 2000: NetBIOS null session not allowed XP Professional: ICF blocked vulnerable ports
Code Red	Removed vulnerable ISAPI filters
Unicode Traversal	Moved web directory to separate drive
SQL Slammer	only use Windows Authentication Mode
Georgi Guninski's security advisory	Internet Explorer 6.0 Security Settings: Disabled Active X
MS01-020	Internet Explorer 5.0 Security Settings: Disabled file downloads

5.2 Significance of Research

The results that show the NSA SNAC guides protect from a number of vulnerabilities as well as patches allows administrators time to test out the patches. Some companies make sure Microsoft patches do not interfere with existing software so using the NSA SNAC guides will help protect their computer systems during this validation process time.

5.3 Recommendations for Action

While the NSA SNAC guides alone provide a better protection against just patches, it is not the intention of this experiment to persuade anyone to stop using patches. The NSA SNAC guides also advocate the use of defense in depth. They recommend not only the use of patches, but firewalls, virus scanning software as well as user education. While the NSA SNAC guides protected against all the attacks that are used in these experiments, there is no guarantee that they will protect against all vulnerabilities by themselves. Computers can be best protected against vulnerabilities through constant reevaluations of security practices.

It should be a priority for the NSA to produce non-technical guidelines to secure: Windows XP Home / Professional, and Windows 2000 as well as common applications since home users are now being targeted by many exploits. These non-technical home users need simple and concise checklists in order to be used. The current NSA SNAC guides are in-depth guides written for knowledgeable system administrators. These would frustrate common users and prevent them from being used.

5.4 Summary

This chapter covered the conclusions made from the results of the experiments. While the NSA SNAC guides seem to work as well as the Microsoft patches it is not recommended to use the NSA SNAC guides alone. The real strength of the NSA SNAC guides is that they promote defense in depth and don’t just rely on one method of protection to defend against exploits.

Bibliography

[Abr04] Abrams, T., *Microsoft LSASS Buffer Overflow from exploit to worm.* www.giac.org/practical/GCIH/Travis_Abrams_GCIH.pdf, Apr. 2004.

[Adg03] Adams, J. and F. Guterl, *Bringing Down the Internet*, Newsweek International, vol. 2004; http://msnbc.msn.com/id/3339638, Oct. 28, 2003.

[Arc99] Archambault, J., *The history of worm like programs*, http://www.snowplow.org, Jul. 2001.

[ArR01] Arquilla, J., D. Ronfeldt, *Networks and Netwars: The Future of Terror, Crime, and Militancy*, RAND Corp, http://www.rand.org/publications/MR/MR1382/, May 2001.

[BCH03] Bickel, R., M. Cook, J. Haney, et al., *Guide to Securing Microsoft Windows XP*, http://www.nsa.gov/snac/os/winxp/winxp.pdf, Dec. 2003.

[Bou03] Boutin, P., *Slammed! An inside view of the worm that crashed the Internet in 15 minutes*. http://www.wired.com/wired/archive/11.07/slammer_pr.html, Jul. 2003.

[CER01] CERT/CC, *CERT Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow in IIS Indexing Service DLL*, http://www.cert.org/advisories/CA-2001-19.html, Jan. 2002.

[CER01a] CERT/CC, *CERT Advisory CA-2001-26 Nimda Worm*, http://www.cert.org/advisories/CA-2001-26.html, Sep. 25, 2001.

[CER02] CERT/CC, *CERT Incident Note IN-2001-09*, http://www.cert.org/incident_notes/IN-2001-09.html, Aug. 6, 2001.

[CER03] CERT/CC, *CERT Advisory CA-2003-04 MS-SQL Server Worm*, http://www.cert.org/advisories/CA-2003-04.html, Jan. 27, 2003.

[ChH00] Christman, S. and J. Hayes, *Guide to Secure Configuration and Administration of Microsoft SQL Server 2000*, http://www.nsa.gov/snac/db/mssql_2k.pdf, Aug. 2003.

[Chr01] Christman, S., *Guide to Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services*, http://www.nsa.gov/snac/os/win2k/w2k_cert_services.pdf, Oct. 10, 2001.

[Chr01a] Christman, S., *Guide to Secure Configuration and Administration of Microsoft Windows 2000 Certificate Services (Checklist Format),* http://www.nsa.gov/snac/os/win2k/w2k_cert_services_checklist.pdf, Oct. 10, 2001.

[Cor01] Corrie, J., *Federal Systems Level Guidance for Securing Information Systems,* http://www.sans.org/rr/whitepapers/policyissues/489.php, Aug. 16, 2001.

[Doe02] Doernberg, C., *Guide to Securing Microsoft Internet Explorer 5.5 Using Group Policy*, http://www.nsa.gov/snac/os/winxp/winxp.pdf, Jul. 2002.

[Fre03] Freeman, W., *An Analysis of the Microsoft RPC/DCOM Vulnerability*, http://www.giac.org/practical/GCIH/Wayne_Freeman_GCIH.pdf, Sep. 22, 2003.

[Gun01] Guninski, G., *MS Office XP – the more money I give to Microsoft, the more vulnerable my Windows Computers are*, http://www.guninski.com/vv2xp.html, Jul. 2001

[HaM02] Haney, J. and O. McGovern, *Guide to Securing Microsoft Windows 2000 File and Disk Resources*, http://www.nsa.gov/snac/os/win2k/w2k_file_disk_resource.pdf, Nov. 2002.

[Han01] Haney, J., *Guide to Securing Microsoft Windows 2000 Group Policy*, http://www.nsa.gov/snac/os/win2k/w2k_group_policy.pdf, Sep. 13, 2001.

[Han01a] Haney, J., *Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set*, http://www.nsa.gov/snac/os/win2k/w2k_group_policy_toolset.pdf, Dec. 1, 2003.

[Huc03] Huckaby, T. *Web Server Market Share and HTTP Compression*, http://www.windowsitpro.com/Windows/Article/ArticleID/39729/39729.html, Jul. 29, 2003.

[Ins01] Inside Security, *Microsoft Internet Explorer MIME Header Attachment Execution Vulnerability*, http://www.inside-security.de/msie_mime_demo.html, 2001.

[Lie03] Lieberman, P., *Cratering: Survive and Prevent virus outbreaks*, http://www.lanicu.com/index.cfm/whitepapers/Cratering_Survive_and_Prevent_Virus_Outbreaks?id=450E141831BFF9AFBFD216D57277FB1D, Aug. 28, 2003.

[Lis03] Liston, T., *Worm and Virus Defense: How We Can Protect the Nation's Computers from These Threats Today*. http://www.hackbusters.net/Worm%20and%20Virus%20Defense.pdf, Sep. 22, 2003.

[LMK04] Lockwood, J., J. Moscola, M. Kulig, et al., *Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware* http://www.arl.wustl.edu/~lockwood/publications/MAPLD_2003_e10_lockwood_p.pdf, Sep. 11, 2003.

[Mic01] Microsoft, *Microsoft Security Bulletin (MS01-020)* http://www.microsoft.com/technet/security/bulletin/MS01-020.mspx, Mar. 29, 2001.

[Mic02] Microsoft, *The 60 Minute Network Security Guide*, http://www.nsa.gov/snac/support/sixty_minutes.pdf, Jul. 12, 2002.

[Mic03] Microsoft, *Microsoft Security Bulletin MS03-026* http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx, Jul. 16, 2003.

[Mic04] Microsoft, *Microsoft Security Bulletin MS04-011*, http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx, Apr. 13, 2004.

[MPS03] Moore, D., V. Paxson, S. Savage, et al., *Inside the Slammer worm*," Security & Privacy Magazine, IEEE, Vol. 1, pp. 33-39, Jul.-Aug. 2003.

[Naz01] Nazario, J., J. Anderson, R. Wash, et al., *The Future Of Internet Worms*, Crimelabs Research., http://www.crimelabs.net/docs/worms/worm.pdf, Jul. 20, 2001.

[Naz04] Nazario, J., *Defense and Detection Strategies against Internet Worms*, Artech House, INC, Norwood, MA, 2004.

[NSA02] NSA SNAC, *The 60 Minute Network Security Guide (First Steps Towards a Secure Network Environment)*, http://www.nsa.gov/snac/support/sixty_minutes.pdf, Jul. 12, 2002.

[NSA04] NSA SNAC, *Security Configuration Guides Overview*, http://www.nsa.gov/snac/, Apr. 1, 2004.

[Pit03] Pitsenbarger, T., *Guide to the Secure Configuration and Administration of Microsoft Exchange 2000 Version 1.2*, http://www.nsa.gov/snac/os/win2k/exch_2k_v1_2.pdf, Oct. 2003.

[Rob04] Robbins, A., *The Virus Wars*, PC Magazine, pp. 115-118, Jul. 2004.

[SaR00] Sanderson, M., D. Rice, *Guide to Securing Microsoft Windows 2000 Active Directory*, http://www.nsa.gov/snac/os/win2k/w2k_active_dir.pdf, Dec. 2000.

[Sec00] Security Focus, *Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability*, http://www.securityfocus.com/bid/1806/, Oct. 17, 2000.

[Sec03] Security Focus, *Microsoft Windows DCOM RPC Interface Buffer Overrun Vulnerability*, http://www.securityfocus.com/bid/8205, Jul. 16, 2003.

[Sec04] Security Focus, *Microsoft Windows LSASS Buffer Overrun Vulnerability*, http://www.securityfocus.com/bid/10108, Apr. 13, 2004.

[Ste01] Stephens, R., *Guide to Securing Microsoft Windows 2000 DNS*, http://www.nsa.gov/snac/os/win2k/w2k_active_dir.pdf, Apr. 2001.

[Sul98] Sullivan, B., *Remembering the Net Crash of '88*, http://www.msnbc.com/news/209745.asp?cp1=1, Nov. 2, 1998.

[Sym04] Symantec, *W32.Welchia.Worm*, http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html, Feb. 26, 2004.

[Thu03] Thurrott, P., *OS Market Share: Microsoft Stomps the Competition*, http://www.winnetmag.com/Article/ArticleID/40481/40481.html, Oct. 9, 2003.

[Tod03] Todd, M., *Worms as Attack Vectors: Theory, Threats, And Defenses*, http://sans.org/rr/papers/index.php?id=930, Jan. 31, 2003.

[Wal02] Walker, W., *Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0*, http://www.nsa.gov/snac/os/win2k/iis_5.pdf, Mar. 4, 2002.

[Wol03] Wolf, D., *Cyber security. Getting it Right* http://www.nsa.gov/ia/Wolf_SFR_22_July_2003.pdf, Jul. 22, 2003.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 21-03-2005	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* April 2003 – March 2005

4. TITLE AND SUBTITLE **National Security Agency (NSA) Systems and Network Attack Center (SNAC) Security Guides versus Known Worms**	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Sullivan, Matthew W., 2d Lt, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GIA/ENG/05-07
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Harley Parkes, CISSP Chief, Operational Network Evaluations National Security Agency Fort George G. Meade, Maryland 20755-6000 (410) 854-6529	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Internet worms impact Internet security around the world even though there are many defenses to prevent the damage they inflict. The National Security Agency (NSA) Systems and Network Attack Center (SNAC) publishes in-depth configuration guides to protect networks from intrusion; however, the effectiveness of these guides in preventing the spread of worms hasn't been studied.

This thesis establishes how well the NSA SNAC guides protect against various worms and exploits compared to Microsoft patches alone. It also identifies the aspects of the configuration guidance that is most effective in the absence of patches and updates, against network worm and e-mail virus attacks. The results from this thesis show that the Microsoft patches and the NSA SNAC guides protect against all worms and exploits tested. The main difference is NSA SNAC guides protected as soon as they were applied where as the Microsoft patches needed to be written, distributed and applied in order to work. The NSA SNAC guides also provided protection by changing default permissions and passwords some worms and exploits use to exploit the computer as well as removed extraneous packages that could have undiscovered exploits.

15. SUBJECT TERMS

Computer security, Computer viruses, Computer worms, NSA SNAC guides

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Rusty O. Baldwin	
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4445 rbaldwin@afit.edu	
U	U	U	UU	60		